

Amendments to the Claims

Claims 1 - 2 (canceled)

1 Claim 3 (currently amended): The method according to Claim [[1]] 32, wherein the input
2 document is a structured document.

1 Claim 4 (previously presented): The method according to Claim 3, wherein the structured
2 document is encoded in Extensible Markup Language (“XML”).

1 Claim 5 (currently amended): The method according to Claim 1, wherein the generated output
2 comprises at least one object representation generated from the input document.

Claim 6 (canceled)

1 Claim 7 (currently amended): The method according to Claim [[1]] 33, wherein the second
2 syntax level is requested by ~~requesting step further comprises the step of specifying a schema~~
3 name of a schema to which the generated output must adhere.

1 Claim 8 (currently amended): The method according to Claim [[1]] 33, wherein the second
2 syntax level is requested by ~~requesting step further comprises the step of specifying a schema~~
3 name of a schema to be used by the validating parser when generating the output.

1 Claim 9 (currently amended): The method according to Claim 8, wherein the schema name is
2 specified, by the application program, as a feature on an invocation of the validating parser.

Claim 10 (canceled)

1 Claim 11 (currently amended): The method according to Claim ~~[[1]]~~ 32, wherein the first syntax
2 level ~~used for the validating of the input~~ is specified in the syntax of the input document.

1 Claim 12 (currently amended): The method according to Claim 11, wherein the specification in
2 the syntax of the input document uses a schema location construct in the input document.

1 Claim 13 (currently amended): A computer-implemented method of casting objects, ~~further~~
2 comprising steps of:
3 validating syntax elements of an input, using a validating parser, according to a first syntax
4 level while generating output objects, from the input using the validating parser, according to a
5 second syntax level, wherein the generating further comprises suppressing, by the validating
6 parser, at least one of the validated syntax elements from the generated output objects in order
7 that the generated output objects will be valid according to the second syntax level; and
8 providing the generated output objects, by the validating parser, for use by an application
9 program.

1 Claim 14 (original): The method according to Claim 13, wherein the second syntax level is a less-

2 restrictive version of the first syntax level.

1 Claim 15 (original): The method according to Claim 13, wherein the first syntax level is a more-
2 restrictive definition of the second syntax level.

1 Claim 16 (original): The method according to Claim 13, wherein the first syntax level is an
2 extension of the second syntax level.

1 Claim 17 (previously presented): The method according to Claim 13, wherein the first syntax
2 level represents an extension of the second syntax level.

1 Claim 18 (original): The method according to Claim 13, wherein the first syntax level and the
2 second syntax level are defined using schemas.

1 Claim 19 (original): The method according to Claim 18, wherein the schema that defines the first
2 syntax level is an extension of the schema that defines the second syntax level.

1 Claim 20 (previously presented): The method according to Claim 13, wherein the first syntax
2 level represents a plurality of extensions to the second syntax level.

1 Claim 21 (currently amended): The method according to Claim 13, wherein the generated output
2 objects adhere to a schema that defines the second syntax level.

1 Claim 22 (original): The method according to Claim 13, wherein the input adheres to an extended
2 schema that defines the first syntax level.

1 Claim 23 (previously presented): The method according to Claim 22, wherein the generated
2 output objects adhere to a base schema that is extended by the extended schema.

1 Claim 24 (currently amended): A system for applying abstraction to object markup definitions,
2 ~~further comprising:~~

3 a validating parser usable by a computer;

4 first means for using the validating parser, executing on the computer, to validate syntax
5 elements specified in an input document expressed as an object markup definition, wherein the
6 validation is performed according to a first syntax level ~~which allows the object markup definition~~
7 ~~to be successfully validated~~; and

8 second means for using the validating parser, executing on the computer, to apply
9 abstraction to the object markup definition when generating, from the validated syntax elements,
10 output syntax for at least one output object for use by a computer in an application therefrom
11 program, responsive to the first means, wherein the applying of the abstraction further comprises
12 suppressing, by the validating parser from the generated output syntax, at least one of the
13 validated syntax elements, in order that the generated output syntax of each generated output
14 object will be valid ~~generates the at least one output object according to a second different syntax~~
15 ~~level which would not allow the object markup definition to be successfully validated and wherein~~

16 each of the suppressed syntax elements is valid according to the first syntax level but is not valid
17 according to the second syntax level.

1 Claim 25 (currently amended): The system according to Claim 24, wherein the ~~different~~ second
2 syntax level is requested by [[an]] the application program and wherein the application program
3 then consumes ~~that will consume the~~ at least one ~~of the at least one~~ generated output-object
4 objects.

1 Claim 26 (currently amended): A computer program product for parsing of input, the computer
2 program product embodied on one or more computer-usable media and comprising:

3 computer-readable program code for validating, by a parser, syntax elements of an input
4 document according to a first schema when parsing syntax of the input document; ~~wherein the~~
5 ~~first schema defines a first syntax level that enables content in the input to be successfully~~
6 ~~validated;~~ and

7 computer-readable program code for suppressing, by the parser, at least one of the
8 validated syntax elements when generating one or more output from the parsed syntax of the
9 input document, wherein each of the suppressed syntax elements is valid according to the first
10 schema but is not valid ~~objects~~ according to a second schema for which the output is generated.
11 ~~[[,]] upon parsing the successfully-validated content in the input, wherein the second schema~~
12 ~~defines a second syntax level that does not enable the content in the input to be successfully~~
13 ~~validated.~~

1 Claim 27 (currently amended): The computer program product according to Claim 26, wherein
2 the first schema specifies a first syntax ~~[[level]]~~ that is a more-restrictive version of a second
3 syntax specified by the second~~syntax level~~ schema.

1 Claim 28 (original): The computer program product according to Claim 26, wherein the first
2 schema is defined as an extension of the second schema.

1 Claim 29 (original): The computer program product according to Claim 26, wherein the first
2 schema is defined as an extension of some intermediate schema that extends the second schema.

1 Claim 30 (original): The computer program product according to Claim 26, wherein the second
2 schema is a base schema upon which one or more extensions are based, and wherein the second
3 schema is one of the extensions and is based either directly on the base schema or on an
4 intermediate schema that extends the base schema.

1 Claim 31 (currently amended): A computer-implemented method of providing validation and
2 parsing for clients, comprising steps of:

3 providing a validating parser that enables a client to dynamically select ~~[[an]]~~ a syntax
4 abstraction level for use when generating output from the validating parser;

5 obtaining an input document to be validated and parsed for the client;

6 validating syntax elements of the input document with the provided validating parser,
7 wherein the validation is performed according to a first syntax level to which the syntax elements

8 ~~of associated with syntax specified in the input document~~ adhere; and
9 suppressing at least one of the validated syntax elements when generating output from the
10 input document with the provided validating parser, for use by the client, wherein:
11 the generated output has syntax that conforms to the syntax abstraction level that
12 has been dynamically selected by the client; ~~and wherein~~
13 the syntax abstraction level is a ~~refinement~~ less-restrictive version of the first
14 syntax level; and
15 each of the suppressed syntax elements is valid according to the first syntax level
16 but is not valid according to the syntax abstraction level.

1 Claim 32 (new): A computer-implemented method of applying abstraction by a validating parser,
2 comprising steps of:

3 using, by a validating parser, a first syntax level for validating syntax elements when
4 parsing syntax of an input document; and
5 omitting, by the validating parser, at least one of the validated syntax elements when
6 generating output from the parsed syntax of the input document, wherein each of the omitted
7 syntax elements is valid according to the first syntax level but is not valid according to a second
8 syntax level for which the output is generated.

1 Claim 33 (new): The method according to Claim 32, wherein the second syntax level is
2 requested, to the validating parser, by an application program for which the output is generated.